Assessing the Quality of a Knowledge Graph via Link Prediction Tasks

Assessing the Quality of a Knowledge Graph via Link Prediction Tasks

RUIQI ZHU, University of Edinburgh, UK
ALAN BUNDY, University of Edinburgh, UK
FANGRONG WANG, University of Edinburgh, UK
XUE LI, University of Edinburgh, UK
KUWABENA NUAMAH, University of Edinburgh, UK
LEI XU, Huawei Ireland Research Centre, Ireland
STEFANO MAUCERI, Huawei Ireland Research Centre, Ireland
J.Z. PAN, University of Edinburgh, UK and Huawei Edinburgh Research Centre, UK

Knowledge Graph (KG) Construction is the prerequisite for all other KG research and applications. Researchers and engineers have proposed various approaches to build KGs for their use cases. However, how can we know whether our constructed KG is good or bad? Is it correct and complete? Is it consistent and robust? In this paper, we propose a method called *LP-Measure* to assess the quality of a KG via a link prediction tasks, without using a gold standard or other human labour. Though theoretically, the LP-Measure can only assess consistency and redundancy, instead of the more desirable correctness and completeness, empirical evidence shows that this measurement method can quantitatively distinguish the good KGs from the bad ones, even in terms of incorrectness and incompleteness. Compared with the most commonly used manual assessment, our LP-Measure is an automated evaluation, which saves time and human labour.

CCS Concepts: • **Information systems** → *Data management systems*; • **Computing methodologies** → **Semantic networks**.

Additional Key Words and Phrases: Knowledge Graph, Link Prediction, Quality Assessment

ACM Reference Format:
Ruiqi Zhu, Alan Bundy, Fangrong Wang, Xue Li, Kuwabena Nuamah, Lei Xu, Stefano Mauceri, and J.Z. Pan. 2023. Assessing the Quality of a Knowledge Graph via Link Prediction Tasks. 1, 1 (November 2023), 10 pages.

1 INTRODUCTION

A Knowledge Graph (KG) is, simply speaking, a set of subject-predicate-object $\langle s, p, o \rangle$ triples, where the subject s and object o represent some individual entities or conceptual classes, while the predicate p asserts the relationship between s and o [29]. A KG is considered to be a simple, structured, yet expressive multi-graph representation of a knowledge domain. The past decade has seen a rise of the use-cases where KGs play a key role in achieving specific tasks, *e.g.* searching among semi-structured heterogeneous data, question answering, recommendation systems, protein classification, and so on [1, 13, 24, 37]. Consequently, the initial imperative step in leveraging the

Authors' addresses: Ruiqi Zhu, University of Edinburgh, UK, ruiqi.zhu@ed.ac.uk; Alan Bundy, University of Edinburgh, UK; Fangrong Wang, University of Edinburgh, UK; Xue Li, University of Edinburgh, UK; Kuwabena Nuamah, University of Edinburgh, UK; Lei Xu, Huawei Ireland Research Centre, Ireland; Stefano Mauceri, Huawei Ireland Research Centre, Ireland; J.Z. Pan, University of Edinburgh, UK and Huawei Edinburgh Research Centre, UK.

XXXX-XXXX/2023/11-ART $15.00
https://doi.org/

advantages of knowledge graphs is to construct a KG of a specific application domain. Unfortunately, these domain-specific KGs are currently not easy to construct. Moreover, after constructing a KG for a specific domain, we often lack a straightforward method of assessing the quality of such newly created graph.

One general method is to estimate the correctness by sampling a small set of triples from the constructed KG and manually checking the precision of these sampled triples [11, 18]. For example, the SymbolicKD [33] project constructed a Commonsense Knowledge Graph (CKG) with 4.38M triples by prompting a large language model, after which they manually checked the acceptance rate of the sampled 1000 triples. This method costs a lot of time and human labour and it can only estimate the correctness but not the completeness of a KG, and even this relies on the expertise of the checkers.

Motivated by the problem described above, in this paper, we consider how to *automatically* assess the quality of a given KG without either gold standards or human labelling. We propose a method called *LP-Measure* which can automatically assess the quality of a KG via the auxiliary link prediction task. Simply speaking, the main idea is to *remove a small part of the KG, and then apply a standard suite of link prediction tools to check how many of the removed triples can be recovered.* We claim that the more triples can be recovered, the more consistent the original KG is, and the more likely that the original KG is of high correctness and completeness.

We can view this measurement from an intuitive perspective of fitting statistical models on datasets: using the same collection of datasets, the statistical methods that produce more accurate predictions are considered to be better methods (e.g., evaluating probabilistic language models on the GLUE benchmark [30]). Conversely, using the same collection of statistical methods, the datasets that produce more accurate models are considered to be better datasets (e.g., higher-quality corpus lead to more powerful language models). Following the same idea, researchers usually evaluate different link prediction methods on some benchmark KGs, and conversely, we can evaluate the quality of KGs using the benchmark link prediction methods.

Unfortunately, our LP-Measure cannot assess the correctness and completeness of KGs. It is more likely to assess aspects such as consistency and robustness. More details on this are in section §6. Yet we shall argue that though correctness and completeness are the most desirable dimensions of quality, there is no way to measure them directly when gold standards are not available, and the best compromise is consistency. A KG of high consistency may likely indicate a KG of high correctness, and a KG of low consistency certainly indicates a KG of low correctness.

We claim that the main contributions of this paper are:

(1) The development of a measurement called LP-Measure to automatically assess the knowledge graph quality. To be more precise, we mainly assess consistency rather than correctness and completeness.
(2) Design of an experiment to empirically show that our LP-Measure is effective, at least in distinguishing the high-quality KGs from the low-quality ones.

2 RELATED WORK

In this section, we provide a brief survey on knowledge graph quality and the methods for its assessment.. We would like to highlight that all of these assessment approaches in our survey require gold standards from human labour. In contrast, our proposed LP-Measure does not.

2.1 Quality of Knowledge Graphs

"Quality" is a broad and vague term, and could be defined in finer-grained dimensions. Zaveri et al. [36] proposed a comprehensive quality assessment framework designed for Linked Data

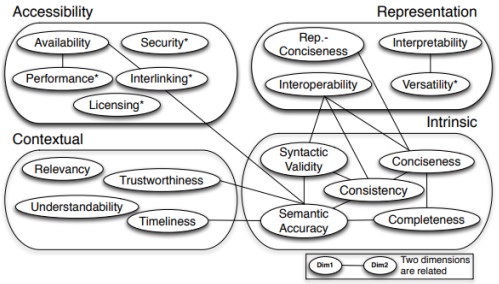


Fig. 1. Linked Data quality dimensions and the relations between them. The dimensions marked with '*' are specific for Linked Data [36]

(LD), which can also be viewed as a knowledge graph but attempts to inter-connect all the open knowledge graphs. The framework refines "quality" to 18 dimensions, grouped into 4 categories, as shown in Figure 1. We can see that some of these quality dimensions are specific to LD, such as whether a KG is licenced, and some are for humans, such as interpretability. In addition, some of them are related, such as consistency, accuracy (correctness), and completeness.

In the field of ontology engineering, a formal domain-independent framework called OntoClean [12] can be used to justify the decisions of ontology engineers without gold standards. This is done by examining the ontology and analysing the meta-properties of ontology classes, such as *identity, unity, rigidity, dependence*, etc.

Different from other datasets in the field of machine learning and general data analysis, Knowledge Graphs (and Semantic Web) usually evolve through time. Most well-known KGs, e.g., DBpedia and Wikidata are updated every day by users around the world. ConceptNet and YAGO also experienced several iterative releases in the past decade. Hence, how well a KG evolves is also an important dimension of quality, and is studied throughly [4, 22].

2.2 Assessing the Intrinsic Quality

Though there are many quality dimensions, we believe that the intrinsic dimensions, including correctness, completeness, and consistency, are more important than others. According to the framework of Zaveri et al [36], correctness (semantic accuracy) means the degree to which the triples can correctly reflect real-world facts, while completeness means the degree to which the entities and relations of a particular domain are represented in a KG (population completeness & property completeness). Consistency means a KG is free of contradictions with respect to its representation and inference mechanism. Generally speaking, consistency is a weaker notion of correctness and completeness. A correct and complete KG should also be consistent, whereas a consistent KG may be a total nonsense.

Most large KGs assess their quality by human evaluation, such as SymbolicKD [33] and Knowledge Vault [10]. As mentioned in §1, this is done by sampling a set of triples and manually checking the correctness of the sampled triples. The obvious drawback of human evaluation is that it costs time and money. There are papers on how to save time and optimise the estimation [11, 18] by designing better sampling strategies, but the drawback still exists.

Some KGs are built to facilitate downstream applications. In this case, the performance of the downstream applications can be used as indirect indicators of the quality of the KGs. A methodology called Competency Questions [8, 23] can be viewed as a general downstream task for ontologies and knowledge graphs. It borrows the idea from test-driven software development: before implementing

any function modules, software engineers clarify the engineering requirement and write the test suites. Then, this test suite can be used as a guide during development, and as an assessment tool after development: the more tests are passed, the higher quality of the software. Competency Questions-driven ontology development still requires the knowledge engineers first to write down a suite of questions that the expected ontology can answer.

There are studies on assessing the quality of triples. For instance, KGTtm [14] proposed a confidence measurement called Triple Trustworthiness to evaluate how much a triple in a KG can be trusted. The Trustworthiness is produced by a neural network trained to capture the triple semantics and global information of the KG. Besides the neural net approach, Inductive Summarisation [3] tried to learn a set of rules that can best summarise the KG, and leverage the induced rules to detect the abnormal triples. Both these methods and other similar approaches can be considered as variants and further applications of the knowledge graph link prediction task, of which the core idea is to train a model that captures the information of the KG, and use the model to score triples (either new triples or existing ones).

3 PRELIMINARY: KNOWLEDGE GRAPH LINK PREDICTION

In this section, we briefly explain some important notions used in our work.

Knowledge Graphs [19] are represented in a standard format for graph-structured data such as RDF. A *knowledge graph* $\mathcal{G}$ is a tuple $(\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a set of relation types, and $\mathcal{T}$ is a set of relational triple $\langle s, p, o \rangle$, Given a set of entities $\mathcal{E}$, relations $\mathcal{R}$, and triples $\mathcal{T}$, a knowledge graph $\mathcal{G} = (\mathcal{E},\mathcal{R},\mathcal{T})$ is a set of tuples, where $s, o \in \mathcal{E}$ are respectively the *head* and *tail* entities of the triple, and $p \in \mathcal{R}$ is the *edge* of the triple connecting head and tail [20].

Ideal Knowledge Graphs [3] is the KG $\mathcal{G}^* = (\mathcal{E}^*, \mathcal{R}^*, \mathcal{T}^*)$ that contains all the correct triples of the domain of interest and no incorrect ones. The notion of correctness (precision) and completeness (recall) is defined by comparing the constructed KG $\mathcal{G}$ with this ideal KG $\mathcal{G}^*$:

$$correctness = \frac{|\mathcal{T} \cap \mathcal{T}^*|}{|\mathcal{T}|} \qquad completeness = \frac{|\mathcal{T} \cap \mathcal{T}^*|}{|\mathcal{T}^*|}$$

Nevertheless, this ideal KG is merely a conceptual aid, and usually does not exist. After all, if we have such an ideal KG at hand, we don't need to bother constructing a KG.

Link Prediction [25, 32], or knowledge completion, is a typical task in the field of Knowledge Graph that aims to predict missing links between entities (triples) of a KG based on its existing knowledge. Our proposed measurement, LP-Measure, relies on knowledge graph link prediction as an auxiliary task.

Commonly used link prediction methods include TransE [6] and ComplEx [28]. Later methods based on deep neural networks include ConvKB [16] and GNN [35]. More advanced methods, based on pretrained language models, include MEM-KGC [7], SimKGC [31], and so forth.

In the link prediction research community, a new prediction model $\mathcal{M}$ is usually evaluated in such a way: take an already known high-quality KG as a benchmark, remove some of its triples, and see how many triples can be recovered (predicted) by the new prediction model $\mathcal{M}$. The more triples that are recovered, the better the performance of the prediction model. For example, when TransE [6] was proposed, it was evaluated in a KG called FB15k, a subset of Freebase [5], and WN11, a subset of WordNet [15].

More concretely, researchers split the triples of a KG into a training set and a test set. The triples in the training set are used to train the prediction model, while the triples in the test set, together with some generated synthetic negative triples are used to evaluate the performance of the trained model. Since almost all KGs only encode positive triples (those they believe to be correct) [2], so one may wonder how we can obtain the negative triples to facilitate training and testing. A

widely-used method in KG link prediction is **negative sampling** under the **Local Closed World Assumption** [17].

To evaluate the performance of a Link Prediction model, we train and then test it: for a positive (correct) triple and several negative (incorrect) triples, the prediction model will output scores of these triples, and rank the triples based on their scores. If more positive triples are ranked ahead of more incorrect ones, then this prediction model is considered to be more powerful in learning the features of a KG and predicting new triples. The commonly used metrics are Mean Reciprocal Rank (MRR) and Hit at K (Hit@k), both of which are within the unit interval $[0, 1]$, the larger the better.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \qquad Hit@k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\mathbb{I}[rank_i < k]$$

where Q is the test set, and $\mathbb{I}[\cdot]$ is the indicator function of an assertion, returning 1 if the assertion is true while 0 if the assertion is false. The removed triples for testing are called *silver standards*[21, 34] instead of gold standards because the benchmark KG, though high-quality, is not perfect.

The evaluation of link prediction algorithms are a kind of controlled experiment: we use the same benchmark KGs (FB15k or WN18), and check the performance of different link prediction models. Conversely, if we use the same benchmark link prediction models, we can also check the quality of different KGs. This is the main idea of our proposed measurement method, LP-Measure.

4 LP-MEASURE

LP-Measure is inspired by the task of link prediction and the idea of controlled experiment. One key observation/assumption of link prediction is that:

> It is possible to predict the missing triples of a KG based on its existing structure. The higher quality a KG is, the more reliably we can predict the missing triples [26, 32].

Here "high quality" means high correctness and completeness, assuming there is an ideal KG. This observation is quite straightforward and intuitive. After all, the task of link prediction is to capture the information and underlying patterns of the existing triples, and leverage them to predict the missing triples. If the existing triples contain a lot of noise or miss a lot of information, then the prediction model cannot learn well and will make inaccurate predictions. Therefore, to be able to conduct the task of link prediction, a precondition is that this KG should be already relatively correct and complete. Taking a further step, we **hypothesise** that

> *For a high-quality KG, we can remove a small part of the KG, and reliably recover (predict) the removed part. The higher the quality is, the more removed triples we can recover.*

In other word, a high-quality KG is also highly recoverable. Based on the hypothesis above, we propose to **use a controlled set of link prediction models** (served as a benchmark) to assess the quality of different KGs.

The idea is simple: first of all, we determine a controlled set of link prediction algorithms as the benchmark, *e.g.*, TransE or ComplEx. Given a KG $\mathcal{G}$ whose quality is unknown, we randomly remove a small part of the KG. We denote the removed triples to be g and the rest to be G (similar to the training/test set split). Then we train the benchmark link prediction model $\mathcal{M}$ on G.

Finally, we apply the trained model to recover the removed triples in g. If most of the removed triples can be recovered, then we can claim that the given KG $\mathcal{G}$ is of high-quality. Algorithm 1 is the pseudo-code of our proposed method. The returned link prediction result, e.g., *mrr* score, indicates the quality of the KG $\mathcal{G}$, the higher the score *mrr*, the higher the quality of $\mathcal{G}$

Algorithm 1: Measuring quality of KG by Link Prediction

Data: The given knowledge graph $\mathcal{G}$, the given link prediction model $\mathcal{M}$
Result: The link prediction result mrr or $hit@k$
$G, g \leftarrow split(\mathcal{G})$;
$m \leftarrow train(\mathcal{M}, G)$;
$mrr, hit@k \leftarrow evaluate\_prediction(m, g)$

5 EXPERIMENT

In this section, we provide empirical evidence to support the effectiveness of our proposed measurement method, LP-Measure. It is a direct application of the hypothesis

> *For a high-quality KG, we can remove a small part of the KG, and reliably recover (predict) the removed part. The higher the quality, the more removed triples we can recover.*

To evaluate the hypothesis, we conducted an empirical experiment hat performed the following:

(1) Identify a well-known good KG $\mathcal{G}$.
(2) Create a worse KG $\mathcal{G}'$ i.e., less correct and less complete ones, based on the good KG $\mathcal{G}$.
(3) Apply LP-Measure on both $\mathcal{G}$ and $\mathcal{G}'$, obtaining corresponding link prediction results. In this experiment, we showed the results of MRR, Hit@1, and Hit@3.

If we see the link prediction results of $\mathcal{G}$ significantly greater than that of $\mathcal{G}'$, then here is the evidence supporting our hypothesis, and justifying the efficacy of our LP-Measure.

Within this framework, we chose 4 good KGs: FB15k, FB15k-237, WN18, WN18RR [6, 9, 27]. These 4 KG datasets are widely used in the link prediction research community, and we believe that the popularity of these benchmark KGs indicates their well-recognised high quality (correctness, completeness, and consistency). We chose 2 models, TransE [6] and ComplEx [28], implemented by Ampligraph[1], which are also well-known in the link prediction community. There are many more powerful and later models, but TransE and ComplEx are simple and easy to run. The deep neural network models or even the transformer models are too slow and demand too much computing power. After all, the important point is to use a fixed link prediction models, not to use the state-of-the-art model. When removing triples, we randomly take off 10% of the triples, i.e., we did a 90%/10% split, and LP-Meausre will check how many triples of the removed 10% can be recovered from the rest 90%.

We created 2 types of "worse" KGs: incorrect KGs and incomplete KGs.

5.1 Incorrect KG

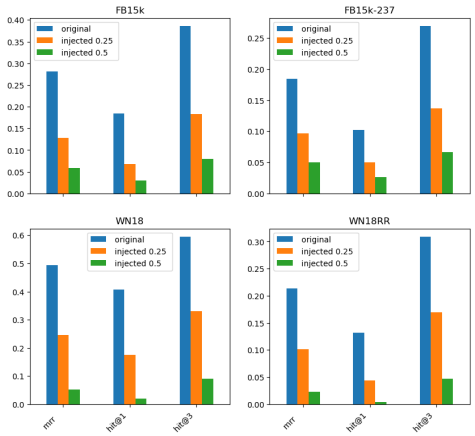
We create an incorrect KG by replacing part of the triples with negative ones generated by negative sampling techniques as mentioned in §3. Here we chose 25% and 50% triples to corrupt because we consider it to be a fair enough proportion to create a noisy KG. We apply the LP-Measure on both good and worse KGs. Figure 2a show the experimental results.

Recall that both MRR and Hit@k are within the unit interval $[0, 1]$, the larger the better. We can see that the link prediction results of the original (good) KGs significantly outperform the injected (worse) KG, which is empirical evidence supporting that our measurement method is effective in distinguishing the correct KGs from the incorrect ones. Given that the more incorrect triples injected, the lower the MRR and Hit@k, we can say that LP-Measure is a quantitative assessment method, i.e., the higher LP-Measure, the better KG quality.

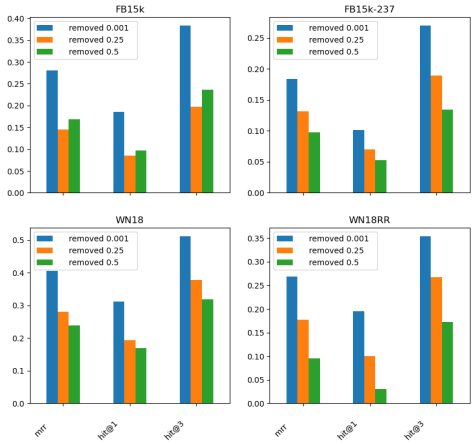
[1] https://github.com/Accenture/AmpliGraph/

5.2 Incomplete KG

We create an incomplete KG by randomly taking out some triples from the original KG. Unlike the previous experiment on incorrectness, we don't injecct any corrupted triples. Again we took out 25% and 50% of the triples, to create incomplete versions compared with the original one. We apply the LP-Measure on both the good and worse KGs. Figure 2b shows the experimental results[2].



(a) LP-Measure results of the original datasets and the worse versions with injected incorrect triples. Metrics are computed by averaging those of TransE and ComplEx.



(b) LP-Measure results of the original datasets and the worse versions with missing triples. Metrics are computed by averaging those of TransE and ComplEx.

Fig. 2

Similar to the previous experiment on inconsistency, We can see that the link prediction results of the original (good) KGs significantly outperform the incomplete (worse) KG, except in the FB15k dataset, where the KG taken out 50% of triples got even better results than the one taken out only 25%. The experimental results of incompleteness do not look as convincing as those of the experiment on incorrectness, this also indicates that the LP-Measure may not be good at distinguishing the complete one and the incomplete one, which is also evidence supporting our thought that the LP-Measure concerns the self-consistency and redundancy of a KG.

Interestingly, the datasets FB15k vs FB15k-237, and WN18 vs WN18RR also constitute 2 pairs of incompleteness comparison. We show the MRR results of them in Table 1. FB15k-237 and WN18RR are subsets of FB15k and WN18 respectively by removing the inverse relations, e.g., `(s, hyponym, o)` and `(o, hypernym, s)` to avoid test leakage [9, 27]. Thus, a link prediction algorithm always gets lower results in FB15k-237 and WN18RR than FB15k and WN18, which is commonsense in the research community. This commonsense is also evidence supporting the claim that the higher LP-Measure a KG would be, the more complete (redundant) it is.

Another interesting result is that the LP-Meausre of WN18 (WN18RR) is significantly better than that of FB15k (FB15k-237). This also aligns with our intuition, since the domain of FB15k is open world knowledge, so we expect that the KG would be fairly incomplete and involve many mistakes. In contrast, the domain of WN18 is merely lexical knowledge, which is a much smaller

[2]We conducted the experiments 3 times and took the average of every datum, so did the later experiments on incompleteness.

	FB15k	FB15K-237	WN18	WN18RR
TransE	0.145	0.127	0.180	0.143
ComplEx	0.417	0.241	0.813	0.395
Average (LP-Measure)	0.281	0.184	0.495	0.269

Table 1. MRRs on all the original datasets

and restricted domain. Therefore, we would expect that such a constructed KG would be more correct and complete than the one of open world knowledge.

6 DISCUSSION AND LIMITATIONS

Though the LP-Measure is simple, effective, and doesn't require gold standards and extra human labelling, is has some limitations.

The most significant limitation is that LP-Measure, by removing part of a KG and measuring how much can be recovered, cannot assess the correctness and completeness of a KG. Rather, it measures the aspect of self-consistency or redundancy of a KG. Since self-consistency and redundancy are weaker notions than correctness and completeness (a correct and complete KG must be self-consistent while a self-consistent KG is not necessarily correct and complete), the LP-Measure is not as powerful and useful as correctness and completeness.

Secondly, it is an indirect measurement. What we can obtain are the link prediction metrics like MRR or Hit@k on that KG, not as straightforward as direct measurements like precision and recall.

Lastly, LP-Measure only works well on large KGs. Because it exploits link prediction models to do the measurement. Most link prediction models require large data to learn the statistical patterns and perform link prediction. Thus, our measurement may not well reflect the quality of small KGs. There could be KGs with less than 100 triples describing a very small domain, e.g., the relationship within a family. In this case, most data-driven link prediction models will not work. A rule of thumb is that, if we can assess the quality of all triples by hand, then we don't need to bother with the LP-Measure.

However, in cases where there are no gold standards to compute precision and recall, our measurement can serve as a quick and easy-to-use probe. We can always use the LP-Measure as a preliminary probe, after which we can decide whether or not to spend some time on human labelling and compute the precision for correctness (and hopefully the recall for completeness).

7 CONCLUSION

In this paper, we proposed a method called the LP-Measure to assess the quality of a KG by means of link prediction tasks. The biggest advantage of our LP-Measure is that it can be run in a fully automated manner, without extra human labour to provide gold standards. Though the applicability is limited, and LP-Measure essentially assesses the consistency of a KG instead of the more desirable correctness and completeness, empirical experiments show that our proposed measurement is effective for large KGs, at least where it can distinguish the good KGs from the bad ones.

ACKNOWLEDGMENTS

The authors would like to thank Huawei for supporting the research on which this paper was based under grant CIENG4721/LSC. We also acknowledge the support of UKRI grant EP/V026607/1, ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence) and EPSRC grant no EP/W002876/1.

Additional appreciation to the anonymous reviewers who offered useful comments for improving the quality of the paper.

REFERENCES

[1] Abdulwahhab Alshammari, Raed Almalki, and Riyad Alshammari. 2021. Developing a Predictive Model of Predicting Appointment No-Show by Using Machine Learning Algorithms. *Journal of Advances in Information Technology* 12, 3 (2021).

[2] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2021. Negative statements considered useful. *Journal of Web Semantics* 71 (2021), 100661.

[3] Caleb Belth, Xinyi Zheng, Jilles Vreeken, and Danai Koutra. 2020. What is normal, what is strange, and what is missing in a knowledge graph: Unified characterization via inductive summarization. In *Proceedings of The Web Conference 2020*. 1115–1126.

[4] Carlos Bobed, Pierre Maillot, Peggy Cellier, and Sébastien Ferré. 2020. Data-driven assessment of structural evolution of RDF graphs. *Semantic Web* 11, 5 (2020), 831–853.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[7] Bonggeun Choi, Daesik Jang, and Youngjoong Ko. 2021. MEM-KGC: Masked entity model for knowledge graph completion with pre-trained language model. *IEEE Access* 9 (2021), 132025–132032.

[8] Matt Dennis, Kees Van Deemter, Daniele Dell'Aglio, and Jeff Z Pan. 2017. Computing authoring tests from competency questions: experimental validation. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16*. Springer, 243–259.

[9] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[10] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 601–610.

[11] Junyang Gao, Xian Li, Yifan Ethan Xu, Bunyamin Sisman, Xin Luna Dong, and Jun Yang. 2019. Efficient Knowledge Graph Accuracy Evaluation. *Proceedings of the VLDB Endowment* 12, 11 (2019).

[12] Nicola Guarino and Christopher Welty. 2002. Evaluating ontological decisions with OntoClean. *Commun. ACM* 45, 2 (2002), 61–65.

[13] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems* 33, 2 (2021), 494–514.

[14] Shengbin Jia, Yang Xiang, Xiaojun Chen, and Kun Wang. 2019. Triple trustworthiness measurement for knowledge graph. In *The World Wide Web Conference*. 2865–2871.

[15] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[16] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2017. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121* (2017).

[17] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2015), 11–33.

[18] Prakhar Ojha and Partha Talukdar. 2017. KGEval: Accuracy estimation of automatically constructed knowledge graphs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1741–1750.

[19] J.Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu (Eds.). 2017. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer.

[20] Jeff Z. Pan. 2009. Resource Description Framework. In *Handbook on Ontologies*. 71–90. https://doi.org/10.1007/978-3-540-92673-3_3

[21] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[22] Mohammad Rashid, Marco Torchiano, Giuseppe Rizzo, Nandana Mihindukulasooriya, and Oscar Corcho. 2019. A quality assessment approach for evolving knowledge bases. *Semantic Web* 10, 2 (2019), 349–383.

[23] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z Pan, Kees Van Deemter, and Robert Stevens. 2014. Towards competency question-driven ontology authoring. In *The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings 11*. Springer, 752–767.

[24] S Revathy, B Bharathi, P Jeyanthi, and M Ramesh. 2019. Chronic kidney disease prediction using machine learning models. *International Journal of Engineering and Advanced Technology* 9, 1 (2019), 6364–6367.

[25] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 2 (2021), 1–49.

[26] Mohammad Javad Saeedizade, Najmeh Torabian, and Behrouz Minaei-Bidgoli. 2022. KGRefiner: Knowledge Graph Refinement for Improving Accuracy of Translational Link Prediction Methods. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 10–16. https://aclanthology.org/2022.sustainlp-1.3

[27] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[29] Boris Villazon-Terrazas, Nuria Garcia-Santa, Yuan Ren, Alessandro Faraotti, Honghan Wu, Yuting Zhao, Guido Vetere, and Jeff Z Pan. 2017. Knowledge graph foundations. In *Exploiting Linked Data and Knowledge Graphs in Large Organisations*. Springer, 17–55.

[30] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 353–355.

[31] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4281–4294.

[32] Meihong Wang, Linling Qiu, and Xiaoli Wang. 2021. A survey on knowledge graph embeddings for link prediction. *Symmetry* 13, 3 (2021), 485.

[33] Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic Knowledge Distillation: from General Language Models to Commonsense Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4602–4625.

[34] Kemas Wiharja, Jeff Z Pan, Martin J Kollingbaum, and Yu Deng. 2020. Schema aware iterative Knowledge Graph completion. *Journal of Web Semantics* 65 (2020), 100616.

[35] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[36] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Soeren Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web* 7, 1 (2016), 63–93.

[37] Xiaohan Zou. 2020. A survey on application of knowledge graph. In *Journal of Physics: Conference Series*, Vol. 1487. IOP Publishing, 012016.